EE 16B Designing Information Devices and Systems II
Summer 2020 S. Kuang, E. Naviasky, F. Laine Project Note 3

Project Phase 3: System Identification

In this phase, we will develop a linear model for the car system and collect data to estimate the free parameters. The specific goals of this phase are as follows:

- Install the remaining sensors
- Verify that all sensors work properly
- Understand the S1XT33N system model
- Collect data
- Use least squares to fit the data to our model and estimate the free parameters

Deriving the Car Model

As we continue developing S1XT33N, we'd like to better understand the car model that we will be using to develop a control scheme. As a wheel on the car turns, there is an encoder disc that also turns as the wheel turns. The encoder shines a light though the encoder disc, and as the wheel turns, the beam of light is interrupted at a rate proportional to the velocity of the car, allowing the encoder to detect how fast the wheel is turning by looking at the number of times that it records an interruption, or a "tick", over a specific time interval.

Our system model is linear, but our system (as with most real-life systems) is not. So, instead of designing a complex nonlinear model, we will approximate the system with a linear model to work for small perturbations around an equilibrium point. You will locate this operating point in part 5 of the lab.

Linear models are particularly useful because their solutions, stability, and controllers can be studied with linear algebra. It is common practice to approximate a nonlinear model with a linear one that is valid near a desired operating point because only limited control methods are applicable to nonlinear systems.

The following model applies separately to each wheel (and associated motor) of the car:

$$v_L[n] = d_L[n+1] - d_L[n] = \theta_L u_L[n] - \beta_L \quad (1)$$

$$v_R[n] = d_R[n+1] - d_R[n] = \theta u_R[n] - \beta_R \quad (2)$$

Meet the variables at play in this model:

- n - The current timestep of the model. Since we model the car as a discrete system, this will advance by 1 on every new sample in the system.
- $d[n]$ - The total number of ticks advanced by a given encoder (the values may differ for the left and right motors–think about when this would be the case).
- $v[n]$ - The discrete-time velocity (in units of ticks/timestep) of the wheel, measured by finding the difference between two subsequent tick counts ($d[n+1] - d[n]$).
- $u[n]$ - The input to the system. The motors that apply force to the wheels are driven by an input voltage signal. This voltage is delivered via a technique known as pulse width modulation (PWM), where the average value of the voltage (which is what the motor is responsive to) is controlled by changing the duty cycle of the voltage waveform. The duty cycle, or percentage of the square wave's period for which the square wave is HIGH, is mapped to the range $[0, 255]$. Thus, $u[n]$ takes a value in $[0, 255]$ representing the duty cycle. For example, when $u[n] = 255$, the duty cycle is 100 %, and the motor controller just delivers a constant signal at the system's HIGH voltage, delivering the maximum possible power to the motor. When $u[n] = 0$, the duty cycle is 0 %, and the motor controller delivers 0 V.
- θ - Relates change in input to change in velocity: if the wheel rotates through n ticks in one timestep for a given $u[n]$ and m ticks in one timestep for an input of $u[n]+1$, then $\theta = m - n = \frac{\Delta v[n]}{\Delta u[n]} = \frac{v_{u_1[n]}[n] - v_{u_0[n]}[n]}{u_1[n] - u_0[n]}$. **Its units are ticks/(timestep $\cdot$ duty cycle).** Since our model is linear, we assume that θ is the same for every unit increase in $u[n]$. This is empirically measured using the car: θ depends on many physical phenomena, so for

the purpose of this class, we will not attempt to create a mathematical model based on the actual physics. However, you can conceptualize θ as a "sensitivity factor", representing the idiosyncratic response of your wheel and motor to a change in power (you will have a separate θ for your left and your right wheel).

- β - Similarly to θ, β is dependent upon many physical phenomena, so we will empirically determine it using the car. β represents a constant offset in velocity, and hence **its units are ticks/timestep**. Note that you will also have a different β for your left and your right wheel.

Linear Least Squares

Before trying to control SIXT33N, we will first determine the system operating point: since your motors are not identical, we need to find an operating velocity that both motors can reach. We will assume that velocity varies approximately linearly with applied voltage, so we will collect data across a range of applied voltages and then perform a least-squares linear regression on a subset of the data (located around your operating point).

In order for the car to drive straight, the wheels must be moving at the same velocity. However, the motors (and hence the wheels) have different achievable velocity ranges, so we need to set the operating point to a velocity achievable by both wheels. A good choice of target velocity is the midpoint of the overlapping range of velocity.

In order to find θ and β, you will perform least squares linear regression on the data you collect in part 4 of the lab.

Linear Regression

The goal of regression is to fit a mathematical system model to a set of observed data points. When we say "fit," what we mean is we want to determine the values of the free model parameters (in our case, θ and β) that allow the model to best approximate the real system performance.

Recall our linear model:

$$v_L[n] = \theta_L u_L[n] - \beta_L$$
$$v_R[n] = \theta u_R[n] - \beta_R$$

We know $v[n]$ and $u[n]$: $v[n]$ is the measured velocity of the wheel (from the encoder data) and $u[n]$ is the input PWM value. $u[n]$ is multiplied by θ, while $v[n]$ is the result, so we can set up the following equations:

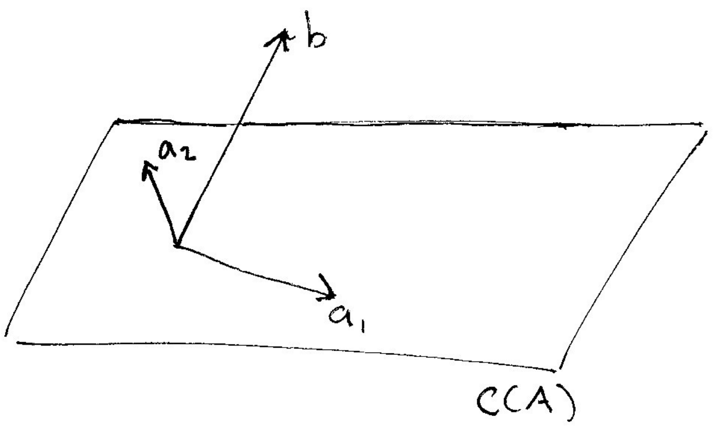
$$\begin{gathered}\theta u[0] - \beta = v[0] \\ \theta u[1] - \beta = v[1] \\ \theta u[2] - \beta = v[2] \\ \vdots \\ \theta u[n] - \beta = v[n]\end{gathered}$$

Now, we can condense the above linear equations into a single matrix-vector equation of the form $Ax = b$ as follows:

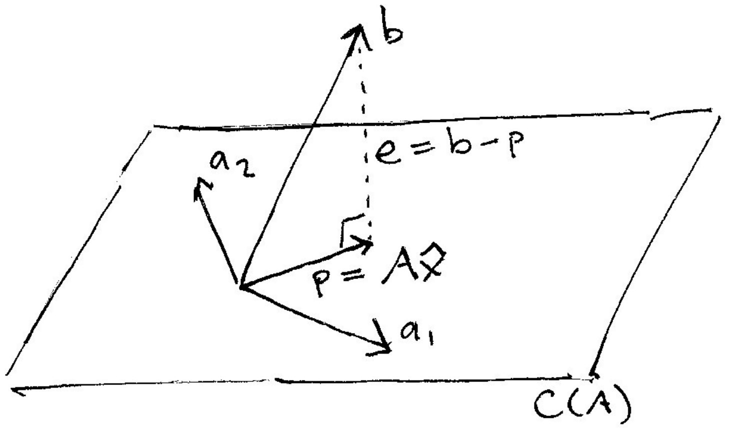
$$\begin{array}{cccc} A & \hat{x} & = & b \\ \begin{bmatrix} u[0] & -1 \\ u[1] & -1 \\ u[2] & -1 \\ \vdots & \vdots \\ u[n] & -1 \end{bmatrix} & \begin{bmatrix} \theta \\ \beta \end{bmatrix} & = & \begin{bmatrix} v[0] \\ v[1] \\ v[2] \\ \vdots \\ v[n] \end{bmatrix} \end{array}$$

Recall that the column space $C(A)$ of a matrix A is the set of linear combinations of the columns of A, and the system $Ax = b$ is solvable only if b (the velocity vector) is in $C(A)$. So, in the equation above, we are asserting that the b vector is in the column space of A, i.e., we're hoping that there's some linear combination of the columns of A that gives us our vector of observed velocity values. However, we already know that b probably doesn't fit the model perfectly, so it's probably not in the column space.

Let's examine this problem graphically. Consider a_1 and a_2 to be vectors in $C(A)$. b is not in $C(A)$, so it does not lie in the plane. The plane $C(A)$ represents the set of velocity vectors the wheel should reach, according to our model.



This is where the regression comes in. Rather than perfectly fitting the observation vector, we want to choose parameters θ and β so that the model outputs the closest possible vector to b that is in $C(A)$ for the given input vector. This vector is the **projection** p of b onto $C(A)$, and $e = b - p$ is the error between the model and the observation .



p is clearly in the column space of A, so it must be equal to A times some vector $\hat{x}$. We want to find the $\hat{x}$ that yields $A\hat{x} = p$. $\hat{x}$ is the vector of θ and β in the matrix equation above, so what we're really saying is we want to find the θ and β for our model that best approximate our observations.

Since e is perpendicular to $C(A)$, we know the dot product $A^T e = 0$. Since $e = b - p$ and $p = A\hat{x}$, we have

$$A^T e = A^T(b - A\hat{x}) = 0$$

and, solving for the parameter vector $\hat{x}$, we have:

$$A^T b - A^T(A\hat{x}) = 0$$
$$A^T b = A^T(A\hat{x})$$
$$\hat{x} = (A^T A)^{-1} A^T b$$

Now you are ready to complete the lab! Go to the jupyter notebook and complete parts 4 and 5 of the lab.

References

Chamberlain, Andrew. (2016) *The Linear Algebra View of Least-Squares Regression*. [online] Available at: https://medium.com/@andrew.chamberlain/the-linear-algebra-view-of-least-squares-regression-f67044b7f39b [Accessed October 14, 2019].

Notes written by Mia Mirkovic (2019)